On degeneracy and collapsing in the construction of the set of objective values in a multiple objective linear program

Jerald P. Dauer

Department of Mathematics, University of Tennessee at Chattanooga, Chattanooga, TN 37403, USA

In this paper an algorithm is developed to generate all nondominated extreme points and edges of the set of objective values of a multiple objective linear program. The approach uses simplex tableaux but avoids generating unnecessary extreme points or bases of extreme points. The procedure is based on, and improves, an algorithm Dauer and Liu developed for this problem. Essential to this approach is the work of Gal and Kruse on the neighborhood problem of determining all extreme points of a convex polytope that are adjacent to a given (degenerate) extreme point of the set. The algorithm will incorporate Gal's degeneracy graph approach to the neighborhood problem with Dauer's objective space analysis of multiple objective linear programs.

Keywords: Multiple objective linear program, degeneracy, collapsing, objective space analysis.

1. Introduction

The underlying program of this work is the multiple objective linear program

$$\text{(MOLP)} \qquad \text{maximize } Cx, \quad x \in X.$$

Here C is a $k \times n$ matrix, and the constraint set X is defined by

$$X = \{x \in \mathbb{R}^n : Ax = b, x \geqq 0\},$$

where A is an $m \times n$ matrix and b is an m-vector. The set of objective values corresponding to X is denoted $Y = C[X]$.

The purpose of this paper is to develop an algorithm to generate all nondominated extreme points and edges of the set Y. The approach, using simplex tableaux for the system defining X, will avoid generating unnecessary extreme points or bases of extreme points of the set X. It is based on, and improves, the algorithm of Dauer and Liu [1] for generating the set of all nondominated extreme points of the set Y. Essential to this procedure is the work of Gal [2] and Kruse [3] on the neighborhood problem, i.e. the problem of determining all extreme points of the convex polytope X that are adjacent to a given (degenerate) extreme point of X. The algorithm will incorporate Gal's [2] degeneracy graph approach to the neighborhood problem with Dauer's [4] objective space analysis of (**MOLP**).

In this algorithm Gal's degeneracy graph approach will be utilized to analyze degenerate extreme points. This will allow the elimination of any analysis of a number of simplex bases that would only contribute redundant information regarding the determination of the edges of the set Y. Also, the linear program Dauer (see Dauer and Liu [1]) developed for determining adjacent nondominated extreme points of Y will be used. This program omits any analysis, other than at most a simple simplex pivot, of extreme points of X that map to the relative interior of faces of Y. The algorithm developed in section 3 will also utilize a framing procedure developed by Wets and Witzgall [5], as was done by Dauer and Liu [1].

The general approach of analyzing the set Y of objective values instead of the constraint set X reduces the number of extreme points of X that need to be considered. The previous objective space work of Dauer [4] on the collapsing effect of the map C from $\mathbb{R}^n$ to $\mathbb{R}^k$ shows that one may expect, at least in a large problem, that a significant number of extreme points of X do not map to extreme points of Y. Therefore, such extreme points will not contribute to the analysis of the problem of generating the set of all nondominated extreme points of the set Y.

For large multiple objective linear programs the computational advantages of the algorithm developed here, which incorporates the Gal–Kruse work on the neighborhood problem, produces significant improvement over the algorithm developed by Dauer and Liu [1]. This paper will also more clearly develop the role that collapsing plays in determining what bases and extreme points of X are necessary to analyze in order to fully determine the edge structure of Y at a given extreme point. The paper by Dauer and Liu [1] fails to express this adequately.

The use of degeneracy graph ideas in solving multiple objective linear programs originally was developed by Gal [2, 6] with a complete treatment given by Kruse [3]. The problem of generating the nondominated extreme points of Y has been of interest in a number of settings. This problem has been of particular interest in the analysis of multiple objective linear programs. Here the work of Dauer [4, 7], Dauer and Liu [1], Dauer and Saleh [8], Gallagher and Saleh [9] all pertain. This problem is also of interest when optimizing a decision function over the set of efficient solutions of a multiple objective linear program. For complete discussions and references for this problem see the work of Benson [10, 11] and Dauer [12].

2. Definitions

For the multiple objective linear program (**MOLP**) a point $x' \in X$ is said to be an *efficient point* (*efficient solution*) *of* (**MOLP**) if there is no point $x \in X$

for which

$$Cx \geqq Cx' \quad \text{and} \quad Cx \neq Cx'.$$

If x' is an efficient point of (**MOLP**), then the corresponding objective value $y' = Cx'$ is said to be a *nondominated* point of Y. The purpose of the algorithm developed in this paper is to generate all of the nondominated extreme points and edges of Y and to do so without generating unnecessary extreme points or bases of extreme points of the set X.

Following the standard terminology of linear programming (e.g., see Luenberger [13]), a basic feasible solution of a linear system $Ax = b$, $x \geqq 0$, is said to be *degenerate* if one or more of the basic variables has zero value. The number of basic variables that are zero for a basic feasible solution is called its *degeneracy degree* (Kruse [3]).

If x is a degenerate basic feasible solution then it has more than one simplex tableau representing it. However, as Gal [2] and Kruse [3, p. 17] have shown, some of these tableaux do not have pivots that will yield distinct, adjacent extreme points of X; such a tableau will be called an *internal* tableau. Others, called *transition* tableaux, possess at least one pivot that will yield a point distinct from x. The internal tableaux yield no information about the structure of Y at $y = Cx$. Thus, it will be advantageous to eliminate analysis of as many such tableaux as possible.

In earlier work Dauer [4, 7] developed an algebraic characterization of the relationship between a face of Y and its corresponding faces of X. Those results demonstrate the collapsing of the structure of X, the flattening of X, to the simpler structure of Y. In particular, Dauer showed that an extreme point of X may be mapped to a point that is not an extreme point of Y, and that in fact an edge of X is not necessarily mapped to an edge of Y. Specifically a face F of X is said to *collapse* (under C) if the dimension of $C[F]$ is less than the dimension of F (see Dauer [4]).

Suppose x^0, x^1 are two extreme points of X. Of interest in this paper is whether or not the entire edge $[x^0, x^1]$ maps to a single extreme point $y^0 = Cx^0 = Cx^1$ of Y. If the edge does collapse to y^0 it is likely that for either extreme point of X the simplex tableaux for that point do not contain all of the information necessary to construct all of the edges of Y at y^0. Thus, it will be necessary to generate and analyze all efficient extreme points of X that map to the extreme point y^0 in order to be able to generate all of the nondominated edges of Y at y^0. However, as noted above, it is not necessary to generate, nor to analyze extreme points of X that do not map to edges of Y. Further, it is necessary to do at most a single pivot at extreme points of X that map to an edge of Y but not to an extreme point. Thus, the objective space approach developed here, which utilizes the Dauer linear program for determining adjacent extreme points of Y, reduces considerably the number of extreme points of X that need analysis. (The computational advantages of the y-space approach resulting from the collapsing effect is well documented; e.g., see Dauer [4, 7, 12, 14] and Dauer and Liu [1].)

One important test that needs to be made during this procedure is to determine if a given edge of Y is nondominated. There are several well-known tests for nondominance (efficiency), see Gal [6] for references.

3. Determining nondominated extreme points

Suppose $\bar{x} \in X$ is an efficient extreme point of X which maps to a nondominated extreme point $\bar{y} = C\bar{x}$ of Y. Such an extreme point can be efficiently generated using the result of Dauer and Liu [1, remark 4.1]. The algorithm developed below determines the nondominated edges of Y that are adjacent to the extreme point $\bar{y}$. The algorithm then uses these edges with the linear program developed by Dauer (see Dauer and Liu [1]) to determine the nondominated extreme points of Y that are adjacent to $\bar{y}$. This algorithm can be used to generate all nondominated extreme points and edges of Y.

First, several ideas from the simplex method will be briefly reviewed. Let $\bar{x}$ be an extreme point of X with a basis matrix B having, without loss of generality, the first m columns of A as basic columns; i.e., suppose that the extended (multiple objective) simplex tableau

$$\begin{bmatrix} A & b \\ C & 0 \end{bmatrix} \gg \begin{bmatrix} I & D & \bar{x}_b \\ 0 & R & \bar{y} \end{bmatrix}$$

at the point $\bar{x}$ has $A = [B, N]$, $D = B^{-1}N$, $\bar{x}_b = B^{-1}b$, $\bar{y} = C(\bar{x}_b, 0)^{\mathrm{T}}$, and $R = C_B B^{-1} N - C_N$. Then R is the matrix of reduced cost coefficients whose columns are the respective changes in the objective value $y = Cx$ for changes in components of the variable x [13, pp. 37–38]; i.e., if i is a nonbasic index for $\bar{x}$ with corresponding column r^i of R, then pivoting on the ith column in the tableau and changing the ith component of $\bar{x}$ to α yields a respective change in the objective value of

$$y = \bar{y} + \alpha r^i. \tag{3.1}$$

Thus, if $\bar{y}$ is an extreme point of Y and if pivoting on the ith column of A corresponds to an edge of X at $\bar{x}$ that maps to an edge of Y with end point $\bar{y}$, then r^i is the direction of that edge of Y and the edge is given by (3.1) for some interval $0 \leqq \alpha \leqq \bar{\alpha}$. The value of $\bar{\alpha}$ can be calculated by the following linear program due to Dauer [1, p. 354]

$$\bar{\alpha} = \text{maximum } \alpha$$

$$\text{subject to } \; Cx - \alpha r^i = \bar{y},$$

$$Ax = b,$$

$$\alpha \geqq 0, \quad x \geqq 0.$$

Thus, if $\bar{\alpha} < \infty$, then $y^1 = \bar{y} + \bar{\alpha} r^i$ is an extreme point of Y adjacent to $\bar{y}$ and this edge of Y is given by $[\bar{y}, y^1]$. It should be noted that this linear program has initial solution $\bar{x}$ and that each pivot will be an extreme point of X that maps to the relative interior of this edge of Y. Thus, these (interior) extreme points of X do not have to be analyzed, other than to be pivoted through in this linear program, in order to determine the structure of Y. This is demonstrated in the following example of Dauer and Liu [1].

EXAMPLE 1

Take C to be a 2×3 matrix and let $x \in \mathbb{R}^3$ have components x_i for $i = 1, 2, 3$. The multiple objective

$$Cx = \begin{bmatrix} 9x_1 + x_3 \\ 9x_2 + x_3 \end{bmatrix}$$

is to be maximized subject to the constraints

$$\begin{gathered} 9x_1 + 9x_2 + 2x_3 \le 81, \\ 8x_1 + x_2 + 8x_3 \le 72, \\ x_1 + 8x_2 + 8x_3 \le 72, \\ 7x_1 + x_2 + x_3 \ge 9, \\ x_1 + 7x_2 + x_3 \ge 9, \\ x_1 + x_2 + 7x_3 \ge 9, \\ x_1 \le 8, \ x_2 \le 8, \\ x_i \ge 0 \quad \text{for all } i. \end{gathered}$$

The seven efficient extreme points of X for this multiple objective linear program are

$$\begin{gathered} x^1 = (0.8, 8.0, 0.9)^{\mathrm{T}}, \\ x^2 = (1.0, 8.0, 0.0)^{\mathrm{T}}, \\ x^3 = (8.0, 1.0, 0.0)^{\mathrm{T}}, \\ x^4 = (8.0, 0.8, 0.9)^{\mathrm{T}}, \\ x^5 = (4.0, 4.0, 4.5)^{\mathrm{T}}, \\ x^6 = (0.0, 8.0, 1.0)^{\mathrm{T}}, \\ x^7 = (8.0, 0.0, 1.0)^{\mathrm{T}}. \end{gathered}$$

Letting $y^i = Cx^i$, for $i = 1, 2, \ldots, 7$, yields that the nondominated extreme points of Y are y^1, y^4, y^6, and y^7. The extreme points x^2, x^3 and x^5 of X map to the relative interior of the edge $[y^1, y^4]$, see fig. 1. The reduced cost matrix at the point x^1 is given by

$$R = \begin{array}{c} \begin{array}{ccc} \overbrace{}^{s^1} & \overbrace{}^{s^3} & \overbrace{}^{s^8} \end{array} \\ \left[\begin{array}{rrr} 0.11270 & -0.01428 & -0.9 \\ -0.00159 & 0.01428 & 0.9 \end{array}\right] \end{array},$$

where s^i, $i = 1, 3, 8$, represent the (nonbasic) slack variables for the corresponding constraints. Either s^3 or s^8 will produce efficient extreme points of X. However, pivoting in s^1 will yield x^2, whereas s^8 will produce x^5. Hence, neither will produce

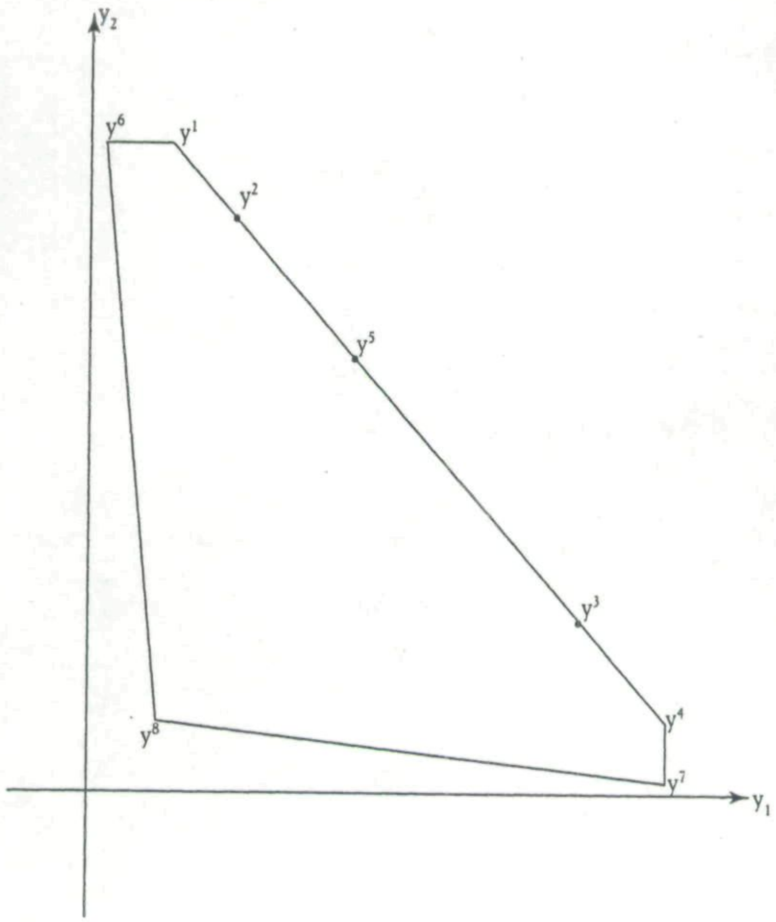

Fig. 1.

the extreme point y^4 of Y directly. Either of these two columns of R can be used in the above linear program in order to produce y^4. □

Several things concerning the algorithm remain to be discussed. First, in the case that the extreme point $\bar{x}$ is degenerate one must determine which simplex tableaux at $\bar{x}$ need to be analyzed in order to determine the edges of Y at $\bar{y}$. Since the edges of Y are determined by a subset of the edges of X, the internal tableaux at $\bar{x}$ are not utilized for the construction of the edges of Y. In fact, in order to determine the corresponding edges of Y at $\bar{y}$ by this approach one only needs sufficient transition tableaux so as to construct all vertices of X adjacent to $\bar{x}$. Determining the minimum number of such tableaux amounts to solving the neighborhood problem at $\bar{x}$. Gal's degeneracy graph technique is used to solve this problem. It should be noted that some internal tableaux may be used in the solution of the neighborhood problem. However, there exists an N-tree method which uses only transition tableaux (see Gal and Geue [17]).

EXAMPLE 2

Consider the following system of constraints developed in Kruse [3, pp. 9, 13–14, 67–69]. An objective function is not given here since it is not involved in this part of the algorithm. (However, one is included in example 3.)

$$\begin{aligned} 4x_2 + x_3 &\leq 4, \\ x_1 + 5x_2 + x_3 &\leq 5, \\ x_1 + 2x_2 \phantom{{}+x_3} &\leq 2, \\ x_1 + x_2 - x_3 &\leq 1, \\ x_1 \geq 0,\ x_2 \geq 0,\ x_3 &\geq 0. \end{aligned}$$

The extreme point $x^0 = (0,1,0)$ is degenerate with seventeen transition and three internal tableaux. (For a complete listing see Kruse [3, p. 14].) The point x^0 has the following six efficient adjacent extreme points of X.

$$\begin{aligned} x^1 &= (0,0,0)^{\mathrm{T}}, \\ x^2 &= (1,0,0)^{\mathrm{T}}, \\ x^3 &= (2,0,1)^{\mathrm{T}}, \\ x^4 &= (2,0,3)^{\mathrm{T}}, \\ x^5 &= (1,0,4)^{\mathrm{T}}, \\ x^6 &= (0,0,4)^{\mathrm{T}}. \end{aligned}$$

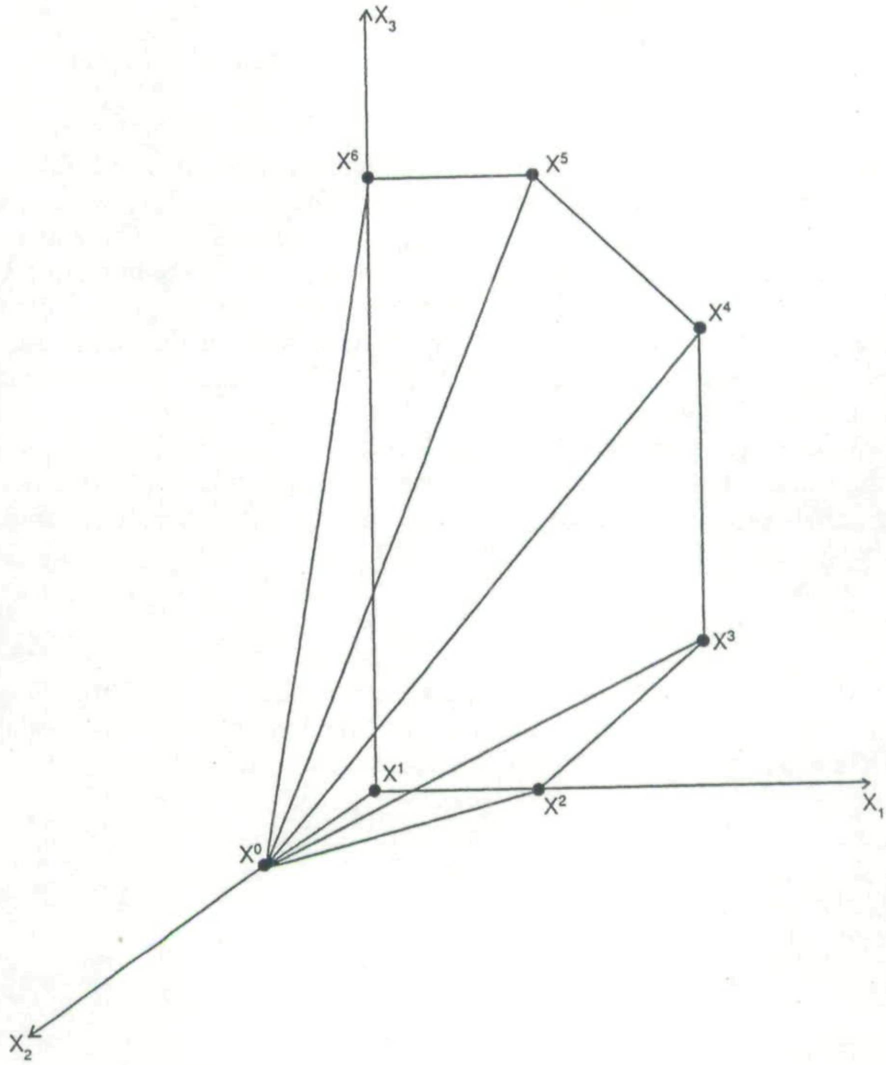

Fig. 2.

A subgraph of the general degeneracy graph which solves the neighborhood problem is given by Kruse [3, p. 69] to be composed of the following four bases

$$B^1 = \{2, 5, 6, 7\},$$

$$B^7 = \{1, 2, 4, 5\},$$

$$B^{15} = \{1, 2, 3, 7\},$$

$$B^{19} = \{1, 2, 5, 7\}.$$

The structure has basis B^{19}, an internal tableau that pivots to the other three bases in this set. The basis B^1 pivots to the adjacent extreme points x^1 and x^6, the basis B^7 pivots to the extreme points x^2 and x^3, and the basis B^{15} pivots to the extreme points x^4 and x^5. The framing set for the point x^0 therefore contains a total of six columns from the reduced cost matrices of B^1, B^7, B^{15} as was identified by solving the neighborhood problem. Each of these corresponds to a unique edge of X which is adjacent to the extreme point x^0. These may or may not correspond to edges of the set Y depending on the coefficient matrix. No further analysis of the internal tableau B^{19} is needed since it contains no information regarding the edges of Y. As is discussed above, once the neighborhood problem has been solved there is no reason to consider analyzing, or even generating, the other tableaux for x^0. □

Vertices of X adjacent to $\bar{x}$ belong to one of two categories. The first category is a vertex x^1 that satisfies $Cx^1 = C\bar{x} = \bar{y}$. By (3.1) both $\bar{x}$ and x^1 map to the same point $\bar{y}$ if and only if the corresponding column r in that simplex tableau for $\bar{x}$ is $r = 0$. In this case the reduced cost matrices for both of these vertices need to be analyzed in order to obtain all of the potential edges of Y at $\bar{y}$.

The second category is an adjacent vertex x^1 that maps to a different point in Y than $\bar{y}$. The simplex pivot corresponding to $\bar{x}$ pivoting to x^1 produces a direction in Y that is possibly an edge of Y. Therefore, the corresponding column of such a simplex tableau of $\bar{x}$, given in the corresponding reduced cost matrix R, needs to be identified. The set of all such directions in Y from reduced cost matrices of vertices of X that map to $\bar{y}$ is called the *framing set* for the extreme point $\bar{y}$ of Y.

In the algorithm the framing set is constructed for an extreme point $\bar{y}$ of Y by an analysis of the necessary transition tableau (i.e., those arising from solving the neighborhood problem) for each vertex of X that maps to the point $\bar{y}$. The Wets–Witzgall [5] algorithm for framing is used to determine a subset of the framing set that will correspond in a one-to-one fashion with the edges of Y at $\bar{y}$.

The framing set aspect of the algorithm below differs from that discussed in the Dauer–Liu paper [1] in two significant ways. The first is that in the Dauer–Liu paper the necessity for determining tableaux from all vertices of X that map to $\bar{y}$ in order to determine the edges of Y is not clear. The second difference is that the framing set in the Dauer–Liu method contains all columns of the all reduced cost matrices at $\bar{x}$. The algorithm developed here does not include the internal tableaux in obtaining the framing set (other than in solving the neighborhood problem). And, as the following comment on example 2 demonstrates, the use of the neighborhood problem may greatly reduce the number of transition tableaux used to construct the framing set.

Note that for the system $Ax = b$, $x \geqq 0$ from example 2 the Dauer–Liu method would present a framing set that contained 80 columns from 20 different reduced cost matrices. Though many of these columns would be duplicates, and easily eliminated, in a large problem the process of framing would be computationally much more complex in the Dauer–Liu algorithm than that using a framing set constructed utilizing the neighborhood problem.

The following algorithm will construct all nondominated extreme points and edges for the set Y of objective values for (**MOLP**). Here the set $\mathscr{A}$ is the set of all nondominated extreme points of Y that have been found but not as yet fully analyzed, $\mathscr{E}$ is the set of all nondominated extreme points of Y found to date, and $\mathscr{G}$ is the set of all nondominated edges of Y analyzed to date.

ALGORITHM

Step 1. Find an initial nondominated extreme point, $y^{(1)} = Cx^{(1)}$, of Y and set $k = 1$, $\mathscr{E} = \mathscr{A} = \{y^{(1)}\}$. (See Dauer and Liu [1, remark 4.1].)

Step 2. At the nondominated extreme point $y^{(k)} = Cx^{(k)} \in \mathscr{A}$, determine a set $\mathscr{N}^k$ of directions that correspond in a one-to-one manner with the nondominated edges of Y at $y^{(k)}$.

Step 3. Set $\mathscr{N} = \mathscr{N}^k \backslash \mathscr{G}$. If $\mathscr{N} = \varnothing$, then there are no (additional) nondominated edges of Y at the point $y^{(k)}$ to be analyzed; set $\mathscr{A} = \mathscr{A} \backslash \{y^{(k)}\}$ and go to step 6.

If $\mathscr{N} \neq \varnothing$, then select $r \in \mathscr{N}$. Set $\mathscr{N} = \mathscr{N} \backslash \{r\}$ and $\mathscr{G} = \mathscr{G} \cup \{r\}$. Go to step 4.

Step 4. Solve the linear program

$$\begin{aligned} &\text{maximize } \alpha \\ &\text{subject to } Cx - \alpha r_i = \bar{y}, \\ &\qquad Ax = b, \\ &\qquad \alpha \geqq 0, \quad x \geqq 0. \end{aligned}$$

Note that $x^{(k)}$ is an initial feasible point to this linear program. The solution of this linear program is an efficient extreme ray or an efficient point $x^{(k+1)}$ of X such that the point $y^{(k+1)} = Cx^{(k+1)}$ is a nondominated extreme point of Y that is adjacent to $y^{(k)}$. (See Dauer and Liu [1].)

Step 5. If $y^{(k+1)} \notin \mathscr{E}$, then set $\mathscr{A} = \mathscr{A} \cup \{y^{(k+1)}\}$, $\mathscr{E} = \mathscr{E} \cup \{y^{(k+1)}\}$, and label $r = [y^{(k)}, y^{(k+1)}] \in \mathscr{G}$ as an edge of Y. Go to step 3 in order to finish the analysis at $y^{(k)}$.

If $y^{(k+1)} \in \mathscr{E}$, then this extreme point of Y has already been found earlier. Go to step 3.

Step 6. If $\mathscr{A} \neq \varnothing$, select $y \in \mathscr{A}$, set $y^{(k)} = y$. Go to step 2. If $\mathscr{A} = \varnothing$, stop.

Remarks on step 2

(1) At the point $y^{(k)}$ we have a simplex tableau corresponding to one basis at $x^{(k)}$. There are several things that must be accomplished at this time based on the

need to determine those nondominated directions in Y at $y^{(k)}$ that correspond to edges of Y. If $x^{(k)}$ is degenerate we must decide which simplex tableaux at $x^{(k)}$ need to be analyzed in order to achieve the goal of determining all of these nondominated edges of Y. The conservative approach proposed by Dauer and Liu [1] was to generate all tableaux at $x^{(k)}$ and take the union of all directions given in these tableaux. But, as discussed above, because of the collapsing effects of C it is necessary to generate all extreme points of X that map to $y^{(k)}$ and generate the tableaux for each of these points in order to obtain the full framing set proposed by Dauer–Liu. As demonstrated by example 2, in many problems the number of tableaux that need to be analyzed can be significantly reduced by using Gal's degeneracy graph ideas and the algorithm of Kruse [13] for solving the neighborhood problem. See Gal [15, 16], Gal and Geue [17], Gal et al. [18] and Kruse [3] for examples and computational information on the number of tableaux and on solving the neighborhood problem.

(2) The question of whether or not a direction is an edge of Y has been addressed by Wets and Witzgall (see [5, section 5] and [19, corollary 7]), though additional research on this question as it applies to the program (**MOLP**) is also being conducted by the author. Wets and Witzgall [5] have developed an algorithm for constructing a frame for a set of vectors. This algorithm can be applied to the framing set at an extreme point of Y in order to generate, in a one-to-one fashion, directions that correspond to the edges of Y at this point.

(3) In general it is not apparent if it is computationally better to reduce the framing set by first determining those directions that are nondominated or if it is more efficient to first determine those directions that are edges of Y. Also, it is not clear which tests for nondominance are more efficient to use in this algorithm. At this time some favor is being given to that developed by Gal [6].

(4) In summary on step 2, one needs to determine which extreme points map to $y^{(k)}$ and then to compute appropriate transition tableaux (as determined from solving the neighborhood problem) for each such extreme point of X. The framing set for $y^{(k)}$ is constructed from the necessary columns from the reduced cost matrices corresponding to these tableaux. This framing set is reduced, in some order, by determining which columns are nondominated directions of Y at $y^{(k)}$ and which of these directions is an edge of Y at $y^{(k)}$. A one-to-one representation of the nondominated edges of Y at $y^{(k)}$ can then be obtained using the Wets–Witzgall [5] algorithm.

EXAMPLE 3

Consider the system $Ax = b$, $x \geqq 0$ defined in example 2 with slack and surplus variables added. Then the constraint set $X \subset \mathbb{R}^7$ has the seven extreme points (see fig. 2)

$$x^0 = (0, 1, 0, 0, 0, 0, 0)^{\mathrm{T}},$$

$$x^1 = (0, 0, 0, 4, 5, 2, 1)^{\mathrm{T}},$$

$$x^2 = (1, 0, 0, 4, 4, 1, 0)^{\mathrm{T}},$$

$$x^3 = (2, 0, 1, 3, 2, 0, 0)^{\mathrm{T}},$$

$$x^4 = (2, 0, 3, 1, 0, 0, 2)^{\mathrm{T}},$$

$$x^5 = (1, 0, 4, 0, 0, 1, 4)^{\mathrm{T}},$$

$$x^6 = (0, 0, 4, 0, 1, 2, 5)^{\mathrm{T}}.$$

Let the matrix C be 3×7 as defined by

$$C = \begin{bmatrix} -1 & 2 & 1 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 & 0 & 1 & 1 \end{bmatrix}.$$

The framing set at $y^0 = Cx^0$ consists of the directions in $Y \subset \mathbb{R}^3$ corresponding to the edges of X given by $[x^0, x^i]$, for $i = 1, 2, \ldots, 6$. These directions are columns in the reduced cost coefficient portions of the tableaux B^1, B^7, B^{15} for the point x^0 in example 2.

For the objective function C the points x^0, x^4, x^5, x^6 are the efficient extreme points of X. Thus, the nondominated edges of Y at y^0 are the edges $[y^0, y^i]$ for $i = 4, 5, 6$. These edges of Y would be added to the set $\mathscr{G}$ by the algorithm, and the extreme point y^0 would be deleted from the set $\mathscr{A}$. The extreme points y^4, y^5, y^6 would be added to the sets $\mathscr{A}$ and $\mathscr{E}$. □

The computational advantages of the y-space approach resulting from the collapsing effect is well documented; e.g., see Dauer [4, 7, 12, 14] and Dauer and Liu [1]. In addition, the advantage of the degeneracy graph approach was discussed in the above, especially in the remarks on step 2. Further, the improvements in the original algorithm of Dauer and Liu [1] by the present algorithm have been described.

The algorithm has been applied with satisfactory results to the multiple objective water resources model of Dauer and Krueger [20]. For this model the number of tableaux computed was about one third less than that used in the Dauer–Liu [1] algorithm. Additional savings, such as those of examples 2 and 3, were not expected for this model because the degenerate extreme points have degeneracy graphs with much fewer nodes than example 2. The Dauer–Krueger model, though of modest size, has been used for comparison of a number of computational techniques and so offers to serve in some capacity as a reliable test model when the degeneracy degree is small.

Other somewhat larger multiperiod, multiobjective models were used to gain additional computational experience with this approach. The results consistently support that using the degeneracy graph algorithm of Kruse at a degenerate extreme point of X results in computing less than 20% of that point's tableaux. The reductions in these results are not as large as in the results of Kruse [3, section 5.2.3.1] where the degeneracy graph approach generally computed less than 9% of the tableaux. However, this difference might well be the result of better programming by Kruse. However, both experiences support that the algorithm developed in this paper is a significant improvement over the method of Dauer–Liu [1].

References

[1] J.P. Dauer and Y.-H. Liu, Solving multiple objective linear programs in objective space, Eur. J. Oper. Res. 46 (1990) 350.

[2] T. Gal, Determination of all neighbours of a degenerate extreme point in polytopes, Discussion paper no. 17b, FernUniversität Hagen, Germany (1978).

[3] H.-J. Kruse, *Degeneracy Graphs and the Neighbourhood Problem* (Springer, New York, 1986).

[4] J.P. Dauer, Analysis of the objective space in multiple objective linear programming, J. Math. Anal. Appl. 126 (1987) 579.

[5] R.J.-B. Wets and C. Witzgall, Algorithms for frames and lineality spaces of cones, J. Res. National Bureau of Standards – B. Math. and Math. Phys. 71B (1967) 1.

[6] T. Gal, A general method for determining the set of all efficient solutions to a linear vector maximum problem, Eur. J. Oper. Res. 1 (1977) 307.

[7] J.P. Dauer, An equivalence result for solutions of multiobjective linear programs, Comp. Oper. Res. 7 (1980) 33.

[8] J.P. Dauer and O.A. Saleh, Constructing the set of efficient objective values in multiple objective linear programs, Eur. J. Oper. Res. 46 (1990) 358.

[9] R.J. Gallagher and O.A. Saleh, A representation of an efficiency equivalent polyhedron for the objective set of a multiple objective linear program, Technical Report, Department of Mathematics, University of Tennessee at Chattanooga, USA (1992).

[10] H.P. Benson, Optimization over the efficient set, J. Math. Anal. Appl. 98 (1984) 562.

[11] H.P. Benson, An all-linear programming relaxation algorithm for optimizing over the efficient set, J. Global Optim. 1 (1991) 83.

[12] J.P. Dauer, Optimization over the efficient set using an active constraint approach, Zeits. Oper. Res. 35 (1991) 185.

[13] D.G. Luenberger, *Introduction to Linear and Nonlinear Programming* (Addison–Wesley, Reading, MA, 1973).

[14] J.P. Dauer, Numerical methods in multicriteria optimization, in: *Multicriteria Optimization in Engineering and in the Sciences*, ed. W. Stadler (Plenum Press, New York, 1988) p. 27.

[15] T. Gal, On the structure of the set bases of a degenerate point, J. Optim. Theory Appl. 45 (1985) 577.

[16] T. Gal, Degeneracy problems in mathematical programming and degeneracy graphs, Oper. Res. Soc. South Africa 6 (1990) 3.

[17] T. Gal and F. Geue, The use of the TNP-rule to solve various degeneracy problems, Oper. Res. Lett. 11 (1992) 23.

[18] T. Gal, H.-J. Kruse and P. Zörnig, Survey of solved and open problems in the degeneracy phenomenon, Math. Progr. 42 (1988) 125.
[19] R.J.-B. Wets and C. Witzgall, Towards an algebraic characterization of convex polyhedral cones, Numer. Mathematik 12 (1968) 134.
[20] J.P. Dauer and R.J. Krueger, A multiobjective optimization model for water resources planning, Appl. Math. Modelling 4 (1980) 171.